Volume 8, No. 3, March – April 2017
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info
ISSN No. 0976-5697

Automatic Fault Tolerant Regular Monitor Software System in Alchemi Grid Middleware

Dr.Rajesh Kumar
Assistant Professor, CSE Department
BGIET, Sangrur, India

Abstract: Fault tolerant resource defence is a critical essential for observes in desktop computational grid. The center of attention of this paper to investigate on Fault Tolerant asset taking care of computational force. Alchemi Desktop grid assets require collecting available computational Power. The Alchemi Desktop grid is a important middleware structure to pull together computational strength by an executor for diverse machines. Breakdown and Error in executor machine can create a problem for desktop grid middleware. Execution level flaws are an exceptionally crucial in the Alchemi desktop grid middleware. This issue has not addressed in Alchemi Desktop Middleware. Alchemi Desktop Grid Middleware provides physical procedure for control executor faults. No automatic system exhibit for execution level deficiencies in Alchemi desktop framework middleware. Today we require computerized technique for consistent and adequate machines in Alchemi Desktop middleware. Our Research effort has Designed, Projected and developed mechanized software system based on regular monitor system. Consistent and Defective Machines can distinguish in shared association. Results demonstrate that automatic software system is Cooperative for Monitor and control on executor faults in Alchemi desktop middleware. Agent can start and cease with help of automated system .Proposed automatic system can perform in Milliseconds. Structure is skilled to sense blemished asset and capable to amend the deficiency in execution machine .It will steady to safeguard level of realistic computational force.

Keywords: Computational, fault Tolerant, Alchemi, middleware; Desktop grid

I. INTRODUCTION

Computing force of a specific machine is not acceptable to ride composite methods or inconveniences. To handle these composite methods, there is need of approach which has capacity to work on Existing current base. Desktop grid is grounded on the major hub, which acknowledges the computational occupation and circulates it in quantities of hubs for result. At the point when partner computational network to an electrical force framework, the grid is semi general and every client demonstrate a part to perform as cogenerater [8] .Resource ease of use perform a vital part in desktop grid. Automatic Fault tolerant courses of action are the need of time. Automated Fault tolerant method will bring the resource convenience up in desktop Grid. Alchemi desktop middleware consists of manager and different Executors. This circumstance is proper for local area network and web [5]. Alchemi middleware can installed in local area network, wherever central hub goes about as supervisor can be related to executions hubs. Desktop framework depends on advancement in utilization of available existing resources. An arrangement of somewhere in the range of thousand PCs can execute the applications that further can execute on quick and expensive supercomputer [3].In terms of spending plan super PC is an excessive amount of exorbitant. It is not promising for each social to purchase the supercomputer. Desktop grid can relate to virtual supercomputer, where incredible sum preparing force can accumulate on specific machine. Various Machines in the system with availability of more processor in relating to applications, which can give the speedier Outcome [9].Most difficult task in grid computing is resource allocation [2]. Colossal measure of Resources with awesome ability are accessible in Institutes, colleges, government, Private (associations).

.

II. RELATED WORK

Azeez et al(2011) says in research paper grid computing with Alchemi middlewares permit submission of requests to execute a Job to Grid, It can run anywhere on the network. Therefore, grid Middlewares serve up as a mediator layer that consent to a consistent and standardized right to use to resources managed in the neighborhood with diverse syntax and access methods. In context of ease of access of various Middleware for Grid implementation with dissimilar features, Research paper focuses on a variety of features that are irregular to Alchemi by taking into concern its Architecture, the Operating system, software demand and restriction that are inborn from its practice.

Chopra (2006) says in master Research work concept of backup manager. Backup manager is based on the heart beating method as well as replication. It is based on fault tolerant method to watch the central Point failure. Crash of the central manager, control will take by back up manager. Grid can start again after failure due to central point. This research work does not provide any solution for execution node failure or any automatic solution.

Fulop (2008) say in research paper that composite computational and visualization algorithms need huge quantity of computational processing power. The computing force of a particular computer is unsatisfactory for execution of such difficult programmes. Usually, Big parallel supercomputers or devoted clusters were used for these types of jobs. A further suitable solution, which is appropriate for grid , is based on the utilization of desktop PCs in a Desktop Grid Computing surroundings.

Das and Sarkar (2012) say in research work in fault tolerance resources in computational grid that a variety of diverse assets of different managerial area are virtually dispersed by dissimilar network in computational grids. So any type of failure can take place at any position of time and job execution in grid can fail. Therefore fault tolerance is an essential and demanding problem in grid computing. Dependability of each grid resources cannot be guaranteed. To create computational grids further efficient and consistent fault tolerant arrangement is compulsory. The objective of this research paper is to assess dissimilar obtainable fault tolerance methods valid in grid computing. This paper presents a state of art in a variety of fault tolerance methods and relative study of current programmes.

Latchoumy and Khader(2011) says in research work survey of fault tolerance of grid computing , the chance of a breakdown is larger than in conventional parallel computing. hence, the error tolerance is an significant property in classify to attain consistency, accessibility and QOS. This Research work, gives a survey of diverse fault tolerance methods, fault administration in dissimilar systems and correlated problems. A fault tolerance examination linked with a variety of failure of resources. It includes process crash, processor breakdown and network fail. The survey gives the interconnected research outcome for fault tolerance in separate functional areas of grid infra-structure and provide the future instructions about fault tolerance methods. This paper is good reference for researchers.

III. REVIEW OF PROBLEM

Alchemi middleware demonstrate a huge part in enclosure of desktop computational grid. Alchemi is .net based network processing structure. It is cooperative to build the desktop based grid in window environment. Alchemi Desktop Grid is made by positioning a director (manager) and executer hubs to append the supervisor [5, 4]. Alchemi middleware grant for asset devouring and asset designation for execution of jobs on the various executing components. Alchemi is open source middleware which can accumulate available force in the system which can see as desktop network. Fault can happen anytime. Running jobs running can fall flat because of fault. Adaptation to internal failure is a huge and motivating issue on the grounds that reliance on independently network asset is not ensured [6].Faults on execution side machines if there should be an occurrence of desktop grid can make troubles, which will specifically impact on accessible force of desktop framework. This Research accentuation on the shortcomings made at executer machines in the event of desktop grid. Pulse identification is rudimentary pieces to assemble Grid framework for more unwavering quality [11].This research exertion center to manage the computational power in a short time. Running Middleware on remote machine can get down in LAN. Adaptation to internal failure strategy comprises of finding of shortcomings in grid assets and offers recuperation to permit calculations [7]. This Research work has outlined and built up a structure .it will control the computational force by sense and correct the flaw in executor. Central Point failure is solved by previously [1].Our Research work points to the executor failure in Alchemi computational grid middleware. Automated Fault Tolerant system is the Requirement of time [10]

IV. DEVELOPMENT OF AUTOMATED FRAMEWORK

Development of Interface is agreeable for examination the genuine state of running desktop middleware. The problem of central Point failure in solved in past research. Our Research effort has designed and Developed Automated Framework based on Regular monitor system for Alchemi desktop middleware for find and corrects the executer faults .It will strong to keep up the level of computational force. Structure is produced on premise of the accompanying steps given beneath.

1. Code for right to utilize the process in remote machines in LAN.
2. Code to affirm the present position of execution.
3. Code for find current status of execution machine. (Mechanized time set is Thirty seconds for revive status of agent).
4. Code for consequently detect the broken agent and correct the shortcoming.
5. Code for location and adjustment blame rapidly (In milliseconds).
6. Code for Design of overall formats for client concern.

V. TESTING & RESULTS OF AUTOMATED FRAMEWORK

After the development of the proposed automatic framework, we have tested the framework in peer to peer node in local area network. By interrupt the execution nodes manually. We observed the reaction of proposed framework. We have observed the reading of response time of framework .Framework is respond to the manual fault control methods. Today, It is very necessary to develope the self organized automatic frameworks. We have designed and developed the Framework in that direction to control the faults in executor nodes. Number of Test cases performed given below

1.Test for view GUI Respond before and after Failure
2.Test for check the Executor status.
3.Test for check respond time of framework.
4.Test for check Refresh system of Framework
5.Test for check Available Cpu Usage on Remote node
6. Test for check Available Memory Usage on Remote node

VI. PSEUDOCODE OF PROPOSED AUTOMATED FRAMEWORK

```
Public Display user class
Initialize btnc1 to zero
Initialize Process start time to zero
Initialize reftime to thirty
Declare global data object
Declare public Display User Function
Initialize component function
Assign global data to object
Private Process List Refresh function
Try
Assign the get processes to process local
For each process p in local
If P Process name is equal to Alchemi executor
Return true
Return false
Catch Exception ee
Show Message box
Return False
Private Kill local Process function
Try
Assign the process get processes to process local
For each process p in local
```

If P Process name is equal to alchemi executor

Access the kill function through P

Return True

Return False

Catch Exception ee

Show Message box

Return False

Private Process List Refresh function String pcname

Assign pc name to string PC name

Assign new connection option function to connoptions

Assign Impersonation level impersonate to connoption impersonation

Set connoption enable Privileges to True

Assign strpcname substring to string uname

Set connoption username is unname

Connoption password is admin

Assign new management scope function to management scope my scope

Access connect function using my scope

Assign new management object searcher function to management object searcher

Assign my scope to searcher scope

For each query object in searcher

If query object is equal to Alchemi executor

Return true

Return false

Private Display Load function

Load network pc function

Private Load network pc function

disuserinfo items clear function

Assign image list 1 to display user info small image list

Assign view small icon to dispuserinfo view

Assign new network browser function to new network nb

For each string pc in nb get network computers function

If btnc1 is equal to 1

Show message box

If pc is equal to object local pc name

If process list refresh1function

Display user info items add pc 0

Else

Display user info items add pc 1

Else

If process list refresh pc

Display user info items add pc 0

Else

Display user info items add pc 1

Private timer1 tick function

Ref_time is equal to ref_time-1

Lblrefresh text is equal to automatically refresh in seconds

If Ref_time is equal to zero

Then load network pc function

Set Reference time is equal to 30

Set btnc1 is equal to zero

Private Remote Execute function

Connection options to new connection options function

options Impersonation is equal to Impersonation Level impersonate

options Authentication is equal to Authentication Level default

options Username is equal to username

options Password is equal to password

options Authority is equal to null

options Enable Privileges is equal to true

ManagementScope scope is equal to new Management Scope

Scope Connect function

Using Management Class process is equal to new Management Class win32 process

process Scope is equal to scope

Process Invoke Method function

Private BtnStartProcess_Click function

Process start time is equal to zero

Timer 2 Interval is equal to one

Timer 2 Enabled is equal to true

object command Line

Set string pc name is equal to dispuserinfo Selected Items

If pc name is equal to object Local Pc Name

Process Start

else

string un is equal to pc name Substring

Remote Execute

Set btnc1 is equal to one

Load network PC function

ref time is equal to thirty

Timer 2 Enabled is equal to false

Show Message box executor start on remote machine in milliseconds

Private btn Refresh_Click function

Load network pc function

Set ref_time is equal to 30

Private kill Process Function

Try

string strPCName is equal to dispuserinfo selected items text

ConnectionOptions connOptions is equal to new ConnectionOptions function

connoptions Impersonation is equal to Impersonation Level Impersonate

connoptions Enable Privileges is equal to true

string uname is equal to strPCName Substring

connoptions Username is equal to uname

connoptions Password is equal to admin

Management Scope my Scope is equal to new Management Scope

my Scope Connect function

Management Object Searcher searcher is equal to new Management Object Searcher

searcher Scope is equal to my Scope

for each Management Object queryObj in searcher get function

queryObj Invoke Method function

Catch Management Exception ee

Private btn Kill Process_Click function

Process_start_time is equal to zero

timer2Interval is equal to one

timer2 Enabled is equal to true

string pcname is equal dispuserinfoSelectedItems text

If pcname is equal to obj LocalPcName

killLocalPrecess function

else

kill Process function

btncl is equal to one

loadNetworkPC function

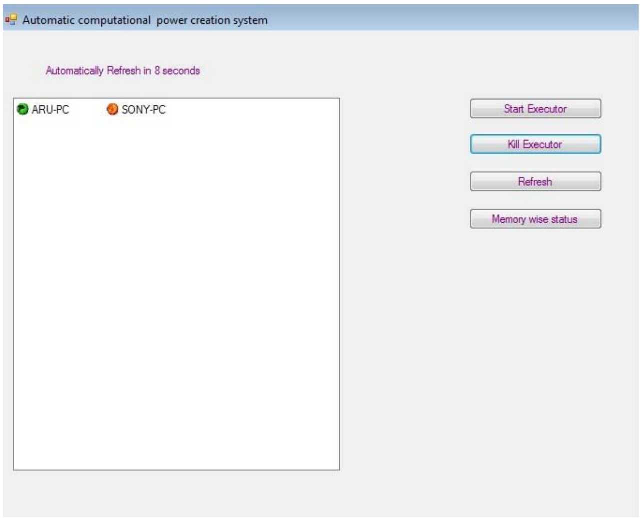
ref_time is equal to thirty

timer2 Enabled is equal to false

show Message Box for Executor stops on Remote Machine in Milliseconds

Memory obj is equal to new Memory

Access show dialog function through object


Automatic computational power creation system

Automatically Refresh in 8 seconds

ARU-PC

SONY-PC

Figure 1. Automated Executor handling GUI system for control the fault

Table I. Response time of Framework for executor fault correction in Alchemi middleware

S.no	Distance Between nodes	Time For Fault correction(milliseconds)
1	2 Meter	87 ms
2	5 Meters	53 ms
3	9 Meters	48 ms
4	12 Meters	62 ms
5	15 Meters	64 ms
6	18 Meters	49 ms
7	22 Meters	55 ms

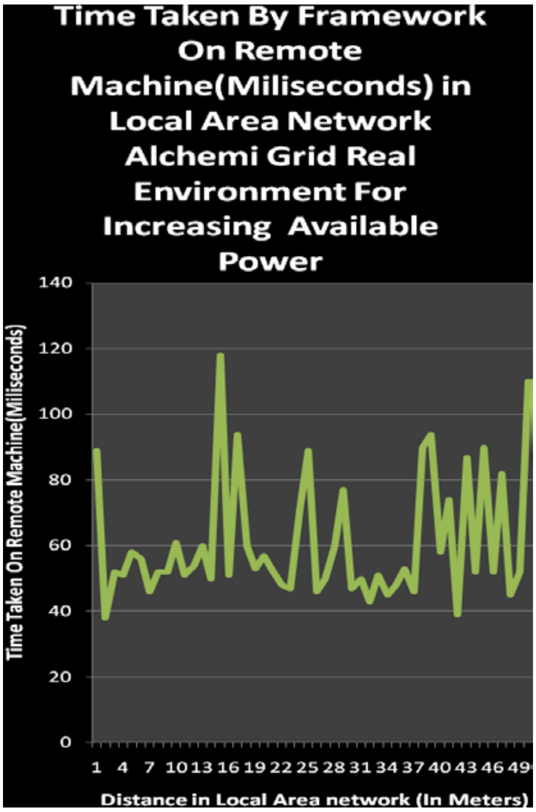
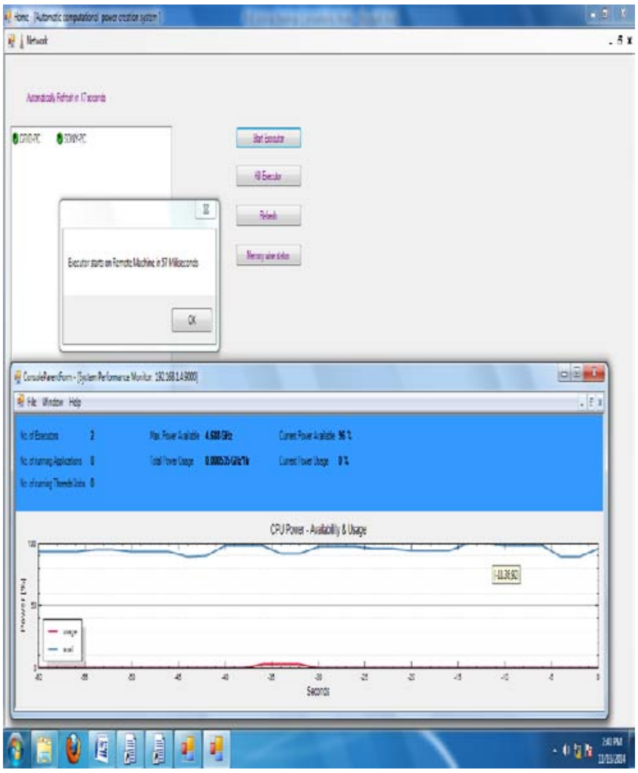


Figure 2-Response Time for Correct Executor fault to maintain the Available computational power



Home - [Automatic computational power creation system]

Automatically Refresh in 17 seconds

Executor starts on Remote Machine in 57 Milliseconds

ConsoleParentForm - [System Performance Monitor: 192.168.1.4:9000]

No of Executors 2 | Max. Power Available 4.680 GHz | Current Power Available 96 %

No of running Applications 0 | Total Power Usage 0.000535 GHz*Hr | Current Power Usage 0 %

No of running Threads/Jobs 0

Figure 3- GUI of Available-Computational Power enhancement by fault correction in Fifty Seven milliseconds

Table II. Response Time Given by Framework Stop Execution Process in Alchemi Desktop Grid Middleware

Sr.no	Distance Between Machines	Time Taken For Terminate Execution in Remote Machine
1	3 Meter	52 ms
2	6 Meters	49 ms
3	9Meters	51 ms
4	15 Meters	66 ms
5	18 Meters	55 ms
6	20 Meters	73 ms
7	24 Meters	143 ms

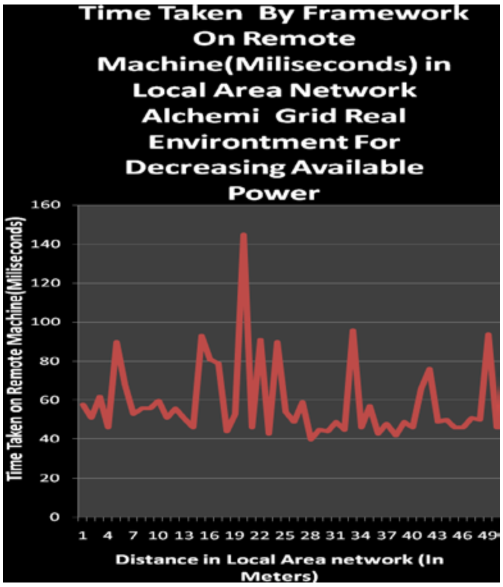
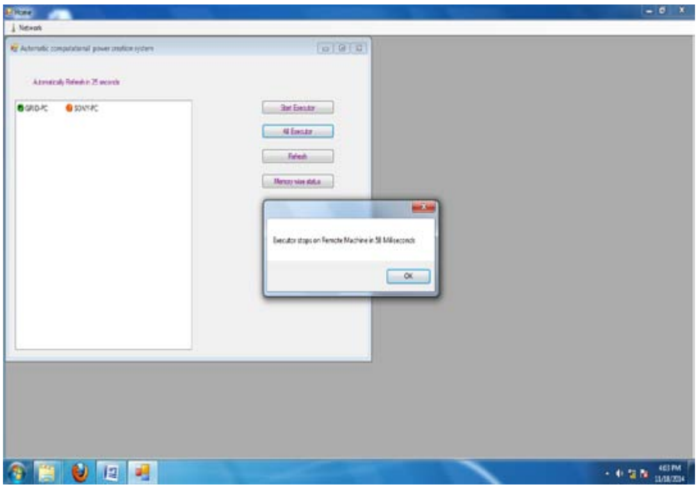


Figure 4- Response Time for Terminate the Executor in Alchemi grid.

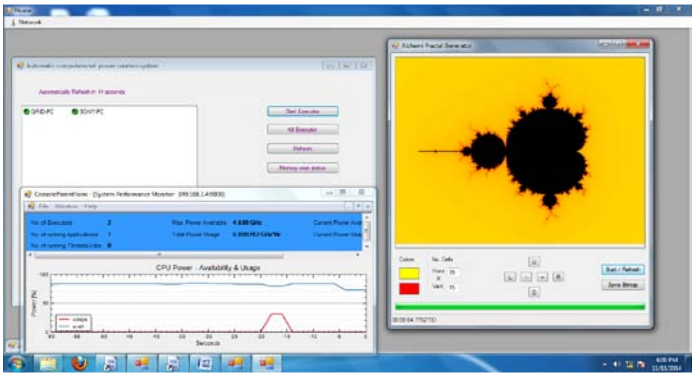


Automatic computational power creation system

Automatically Refresh in 25 seconds

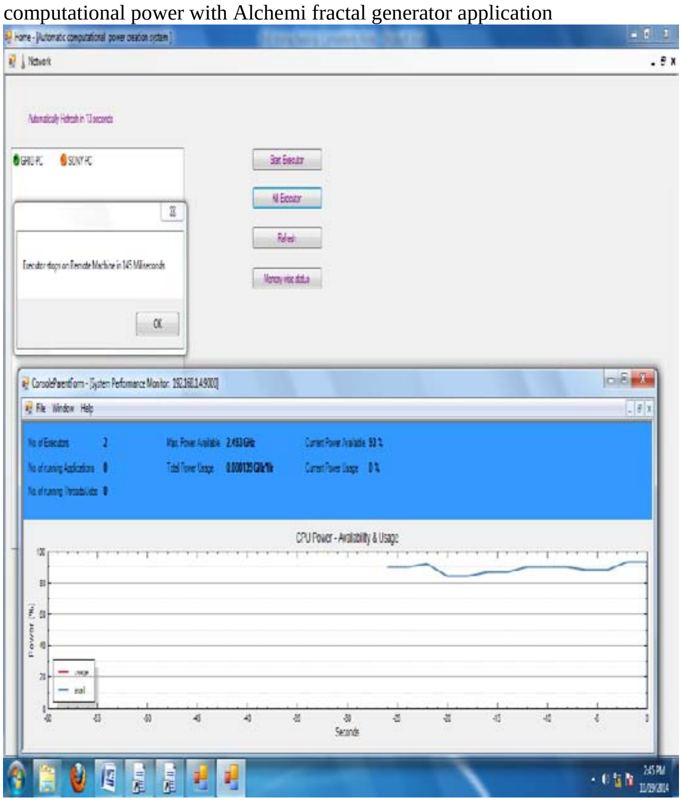
Executor stops on Remote Machine in 58 Miliseconds

Figure 5- Graphical user Interface for automatic termination of Execution process in Fifty Eight Milliseconds



Home
Network
Automatic computational power creation system
Automatically Refresh in 11 seconds
GRID-PC
SONY-PC
Start Executor
All Executor
Refresh
Memory wise status
ConsoleParentForm - [System Performance Monitor: 192.168.1.4:9000]
CPU Power - Availability & Usage
Seconds
Alchemi Fractal Generator
Start / Refresh
Save Bitmap

Figure 6. GUI of Controlling system showing accessibility of Resources for computational power with Alchemi fractal generator application



Home - [Automatic computational power creation system]

Automatically Refresh in 13 seconds

Executor stops on Remote Machine in 145 Milliseconds

ConsoleParentForm - [System Performance Monitor: 192.168.1.4:9000]

No of Executors 2 | Max. Power Available 2.493 GHz | Current Power Available 93 %

No of running Applications 0 | Total Power Usage 0.000125 GHz*Hr | Current Power Usage 0 %

No of running Threads/Jobs 0

Figure 7.Graphical user Interface of Computational Power Control software system showing with console Graphical view

VII. CONCLUSION OF RESEARCH WORK

This Research work has decided on to hold the open handling force in the computational desktop grid middleware, if there should be an occurrence of deficiencies in execution hubs. Exploration consider and built up the structure for protect the computational power in Alchemi grid middleware .This Research work find out most cited inconveniences of Executor fault. Graphical client interface based Framework gave extremely fast and straightforward perspective of results .Alchemi middleware is chosen for the desktop grid cause Alchemi.net grid middleware introduced in Microsoft windows. Our Developed Software framework can control the execution Processes rapidly, which can't deal with by the human manual technique in this quick time. GUI perspective of the Framework is commonsense for the client correspondence with created structure. Programming based created structure for control the reachable computational force is steady for control the Faults. Human endeavors will diminish by use of Software based structure. Manual methodology will crush by use of programmed procedure to oversee Faults in Alchemi desktop middleware. Snappy Control on the procedure on agent machines coordinates in oversees realistic Processing Power. This Research work Conclude that Development of programmed Frame work is basically agreeable for deal with the open Processing Power. Execution machines shortcoming can be control rapidly i.e. in fraction of second. This Research has concluded that Human Efforts in environment of Faults can be therapist by utilize Regular monitor Software system. Manual Method can be removed by use the automatic robotized approach. Today, we require to Develop Fully automatic Software System to control the faults and failures in Grid computing.

VIII. REFERENCES

[1] Chopra I.Fault Tolerence in Computational grid. ME Thesis, Thapar University, Patiala.2006

[2] A.Haruna, Z.Nordin, H.Nazleeni, .Grid Resource allocation, and Research journal of information technology. ISSN: 2041-3114.4(2)pp 38-55

[3] Rajesh K, Surender J.High Computational Power: A Research Challenges in Desktop Grid Computing. In: AEMDS;2013; Elsevier India, isbn-9789351070573, chapter 75, pp 489-492

[4] Akshay L, Rajkumar B, Rajiv R, Srikumar V.Peer-to-Peer Grid Computing and a .NET-based Alchemi Framework.In: High Performance Computing: Paradigm and Infrastructure; june 2005 ; Wiley Press, New Jersey, USA. ISBN: 0-471-65471-X

[5] Akshay L,Rajkumar B, Rajiv R, Srikumar V. Alchemi A .net based grid computing framework and its integration in global grids. Grid Computing and Distributed Systems (GRIDS) Laboratory, Department of Computer Science and Software Engineering, University of Melbourne, Australia.

[6] Arindam D, Ajanta S.on fault tolerance in computational grid. International Journal of Grid Computing & Applications 2012; 3

[7] Christopher D.Reliability in grid computing system. Concurrency and computations:Practice & experience Concurrency Computat.: Pract. Exper.Published Wiley InterScience doi: 10.1002/cpe.1410.2009

[8] Ian F,Kesselman C.Blue Print for a new computing Infrastructure 1998

[9] Monika V, zoran C, Catalina N.Availability of computational Resources for Desktop grid computing. BULETINUL Universită Ńii Petrol – Gaze din Ploiesti 2009; 1: 71-76

[10] Rajesh K,Surender J. Automated Fault Tolerated system for control computational Power in Desktop Grid. In:IEEE International Advance computing Conference; june 2015;banglore india:IEEE,pp 818-821

[11] Zhiming L, jichang S, xiaohua Y.Wan yapping novel dynamic heartbeat detection for grid membership management.IN: IEEE Computer Sciences and Convergence Information Technolog Fourth International Conference; 2009; ISBN: 978-0-7695-3896-9, pp 23-27